Improving Surface Normals Based Action Recognition in Depth Images

Xuan Son Nguyen
Université de Lorraine
Villers-Lès-Nancy, F-54600, France
xuan-son.nguyen@inria.fr

Thanh Phuong Nguyen
Université de Toulon
83957 La Garde, France
thanh-phuong.nguyen@univ-tln.fr

François Charpillet
Inria Nancy - Grand Est
Villers-Lès-Nancy, F-54600, France
francois.charpillet@inria.fr

Abstract

In this paper, we propose a new local descriptor for action recognition in depth images. Our proposed descriptor jointly encodes the shape and motion cues using surface normals in 4D space of depth, time, spatial coordinates and higher-order partial derivatives of depth values along spatial coordinates. In a traditional Bag-of-words (BoW) approach, local descriptors extracted from a depth sequence are encoded to form a global representation of the sequence. In our approach, local descriptors are encoded using Sparse Coding (SC) and Fisher Vector (FV), which have been recently proven effective for action recognition. Action recognition is then simply performed using a linear SVM classifier. Our proposed action descriptor is evaluated on two public benchmark datasets, MSRAction3D and MSRGesture3D. The experimental result shows the effectiveness of the proposed method on both the datasets.

1 Introduction

Approaches for human action recognition in depth images have received a large attention in recent years thanks to the rich information provided by depth sensors.

These approaches usually exploit depth information to build highly discriminative low-level descriptors, or use skeletal data which can be more easily obtained using depth images to build high-level descriptors. Although many approaches have achieved impressive results, they still face a number of challenges, e.g. rate variations, temporal misalignment, composite actions, noise, human-object interaction. Moreover, most of them require high computation time to extract features and recognize actions.

In this paper, we propose a new local descriptor that relies on surface normals in 4D space of depth, time, spatial coordinates and higher-order partial derivatives of depth values along spatial coordinates. The advantage of our proposed descriptor is that it is low-dimensional while being able to jointly capture the shape and motion cues. In order to perform action recognition, we follow the traditional BoW approach and use Sparse Coding [9] and Fisher Vector [19] for obtaining a global representation of depth sequences.

This paper is organized as follows. Section 2 introduces the related work on action recognition in depth images. Section 3 explains our proposed local descriptor and the methods for feature encoding using SC and FV. Section 4 presents the experimental evaluation of the proposed method. Finally, Section 5 offers some conclusions.

2 Related Work

Existing approaches for action recognition in depth images can be broadly grouped into three main categories: skeleton-based, depth map-based and hybrid approaches. Xia et al. [28] partitioned the 3D space using a spherical coordinate defined at the hip joint position. Each 3D joint casted a vote into a bin to generate a histogram of 3D joint. These histograms were then used to construct visual words whose temporal evolutions were modeled using discrete Hidden Markov Models [18]. Yang and Tian [30] learned EigenJoints from differences of joint positions and used Naïve-Bayes-Nearest-Neighbor [1] for action classification. Zanfir et al. [32] relied on the configuration, speed, and acceleration of joints to construct the action descriptor. A modified kNN classifier was then used for action classification. Vemulapalli et al. [22] used rotations and translations to represent 3D geometric relationships of body parts in a Lie group [14], and then employed Dynamic Time Warping [13] and Fourier Temporal Pyramid [25] to model the temporal dynamics. Du et al. [5] divided the human skeleton into five parts, and fed them to five subnets of a recurrent neural network [20]. The representations extracted by the subnets at a layer were hierarchically fused to be the inputs of higher layers. Once the final representations of skeleton sequences have been obtained, actions were classified using

a fully connected layer and a softmax layer.

Depth map-based approaches usually rely on low-level features from the space-time volume of depth sequences to compute action descriptors. Li et al. [11] proposed a bag of 3D points to capture the shape of the human body and used an action graph [10] to model the dynamics of actions. Representative 3D points used to describe a posture were sampled from a very small set of points in depth maps. Kurakin et al. [8] proposed cell occupancy-based and silhouette-based features which were then used with action graphs for gesture recognition. Wang et al. [24] introduced random occupancy patterns which were computed from sub-volumes of the space-time volume of depth sequences with different sizes and at different locations. Yang et al. [31] projected depth maps onto three orthogonal Cartesian planes to obtain Depth Motion Maps (DMMs) which were used to extract HOG descriptors [12] for action recognition. Xia and Aggarwal [27] proposed a filtering method to extract local spatio-temporal interest points of depth sequences. The histograms of depth pixels in 3D cuboids centered around the extracted interest points were calculated and used in a BoW approach for action recognition. Wang et al. [23] represented actions by histograms of spatial-part-sets and temporal-part-sets, where spatial-part-sets are sets of frequently co-occurring spatial configurations of body parts in a single frame, and temporal-part-sets are co-occurring sequences of evolving body parts. Oreifej and Liu [15] and Yang and Tian [29] relied on surface normals in 4D space of depth, time, and spatial coordinates to capture the shape and motion cues in depth sequences. However, they used different methods to construct action descriptors. The method of [15] used polychorons to quantize possible directions of 4D normals, while the method of [29] used SC to compute visual words and spatial average pooling and temporal max pooling to aggregate local descriptors into a global representation of depth sequences. Chen et al. [2] proposed a real-time approach that used local binary patterns [26] computed for overlapped blocks in the DMMs of depth sequences to construct action descriptors. Action recognition was performed using feature-level fusion and decision-level fusion.

Hybrid approaches combine skeletal data and depth maps to create action descriptors. Wang et al. [25] introduced local occupancy patterns computed in spatio-temporal cells around 3D joints which were treated as the depth appearance of these joints. They proposed Actionlet Ensemble Model where each actionlet is a particular conjunction of the features for a subset of 3D joints. An action was then represented as a linear combination of a set of discriminative actionlets which were learned using data mining. Zhu et al. [33] fused spatio-temporal features based on 3D interest point detectors and skeleton joints features using pair-wise joint distances in one frame and joints difference between two consecutive frames.

3 Our Method

3.1 Local Feature Descriptor

Our local descriptor is computed using surface normals in 4D space of depth, time, and spatial coordinates [15, 29]. Those are the extensions of surface normals in 3D space of depth and spatial coordinates [21]. The depth sequence can be considered as a function $\mathbb{R}^3 \rightarrow \mathbb{R}^1 : z = f(x, y, t)$, which constitutes a surface in 4D space represented as a set of points (x, y, z, t) satisfying $S(x, y, t, z) = f(x, y, t) - z = 0$. Denote by $\mathbf{p}_t$ the pixel at frame t of the depth sequence with the spatial coordinates (x, y) and the depth value z. We form a 5-dimensional local descriptor associated with $\mathbf{p}_t$ as follows:

$$\mathbf{l}(\mathbf{p}_t) = [\frac{\partial z}{\partial x}; \frac{\partial z}{\partial y}; \frac{\partial z}{\partial t}; \frac{\partial^2 z}{\partial x^2}; \frac{\partial^2 z}{\partial y^2}].$$

Note that the first three components of $\mathbf{l}(\mathbf{p}_t)$ correspond to the three components of the normal to the surface S at point (x, y, z, t), which have been shown [15, 29] to capture important shape and motion cues.

Denote by $\{\mathbf{p}_{t-\frac{n-1}{2}}, \mathbf{p}_{t-\frac{n-1}{2}+1}, \ldots, \mathbf{p}_{t+\frac{n-1}{2}}\}$ the set of n neighboring pixels of $\mathbf{p}_t$ in the temporal dimension, i.e. these pixels have the same spatial coordinates as $\mathbf{p}_t$ but they are at n neighboring frames of frame t. In order to further introduce the local motion cue, the final feature descriptor $\mathbf{v}$ extracted at pixel $\mathbf{p}_t$ is formed by clustering the 5-dimensional descriptors extracted at pixels $\mathbf{p}_{t-\frac{n-1}{2}}, \mathbf{p}_{t-\frac{n-1}{2}+1}, \ldots, \mathbf{p}_{t+\frac{n-1}{2}}$ as follows:

$$\mathbf{v} = [\mathbf{l}(\mathbf{p}_{t-\frac{n-1}{2}}); \mathbf{l}(\mathbf{p}_{t-\frac{n-1}{2}+1}); \ldots; \mathbf{l}(\mathbf{p}_{t+\frac{n-1}{2}})].$$

In our experiments, the number of neighboring pixels n is set to $n = 5$. Our local descriptors are thus 25-dimensional vectors. In comparison with the local descriptors proposed in [29] (termed *polynormals*) which are 81-dimensional vectors, the dimensionality of our local descriptors is much lower than that of polynormals. As we will see in the next section, we use the method proposed in [29] for feature encoding, which results in action descriptors whose dimensions are proportional to the product of the dimension of a local descriptor and the number of visual words in the codebook. Thus, using the same size for the codebook and the same classifier for action recognition, our method is much more efficient than the method of [29]. Moreover, we show in the experimental evaluation that our method also outperforms the method of [29] in terms of recognition accuracy.

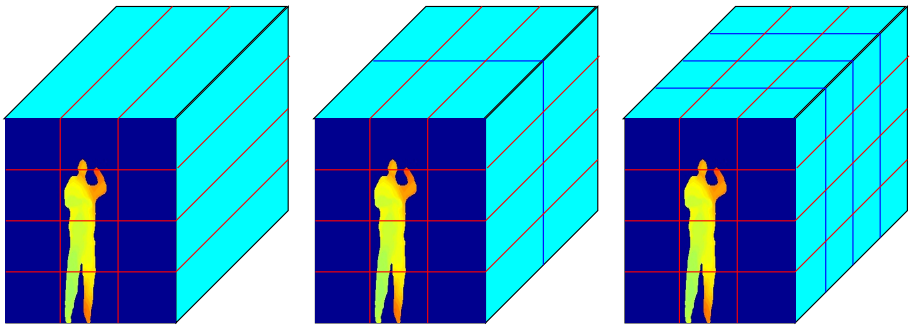

Figure 1: The spatio-temporal grids used in our experiments. From left to right: the first, second and third temporal pyramids

3.2 Feature Encoding

Suppose that for each sequence, we obtain a set of whitened vectors $\mathbf{X} = \{\mathbf{v}_t, t = 1, \ldots, T\}$, where $\mathbf{v}_t \in \mathbb{R}^M$ is a local descriptor, M is the dimension of a local descriptor, T is the number of local descriptors extracted in the sequence. In order to construct a global representation of the sequence, we use SC and FV which are explained in more detail in the next sections.

3.2.1 Sparse Coding

Yang and Tian [29] used SC to train a codebook and code local descriptors. In this method, the codebook is computed by solving the following constrained optimization problem:

$$\min_{\mathbf{D},\boldsymbol{\alpha}} \frac{1}{N} \sum_{i=1}^{N} \left(\frac{1}{2} ||\mathbf{v}_i - \mathbf{D}\boldsymbol{\alpha}_i||_2^2 + \lambda ||\boldsymbol{\alpha}||_1 \right),$$
$$\text{subject to } \mathbf{d}_k^T \mathbf{d}_k \leq 1, \forall k = 1, \ldots, K,$$

where $\{\mathbf{v}_1, \ldots, \mathbf{v}_N\}$ is a subset of $\mathbf{X}$, $\mathbf{D}$ in $\mathbb{R}^{M \times K}$ is the codebook, each column $(\mathbf{d}_k)_{k=1}^K$ represents a visual word; $\boldsymbol{\alpha}$ in $\mathbb{R}^{K \times N}$ is the coefficients of sparse decomposition; λ is the sparsity inducing regularizer.

Each column $(\boldsymbol{\alpha}_i)_{i=1}^N$ is l_1-normalized to obtain the soft assignment $\alpha_{k,i}$ of $\mathbf{v}_i$ to the k^{th} visual word. Assuming that the depth sequence is partitioned into different spatio-temporal grids. Then a local descriptor is computed for each spatio-temporal grid and the final descriptor of the depth sequence is the concatenation of the descriptors from all the grids. For each spatio-temporal grid and each visual

word, the average of the coefficient-weighted differences is computed over all 3D points at the same frame of the grid:

$$\mathbf{u}_k(f) = \frac{1}{|N_f|} \sum_{i \in N_f} \alpha_{k,i}(\mathbf{v}_i - \mathbf{d}_k),$$

where N_f is the set of 3D points at frame f of the grid.

If $\mathbf{u}_k(f) = [u_{k,1}(f); \ldots; u_{k,M}(f)]$, then the vector representation $\mathbf{u}_k = [u_{k,1}; \ldots; u_{k,M}]$ of the k^{th} visual word for the grid is computed by taking the maximal value over all the frames of the grid for each dimension:

$$u_{k,i} = \max_{f=1,\ldots,F} u_{k,i}(f), \text{ for } i = 1, \ldots, M,$$

where the frame indices of the grid are supposed to be $1, \ldots, F$.

The vector representation $\mathbf{U}$ of the grid is the concatenation of the vectors $\mathbf{u}_k$:

$$\mathbf{U} = [\mathbf{u}_1; \ldots; \mathbf{u}_K].$$

In our experiments, we used the spatio-temporal grids proposed in [29] and illustrated in Fig. 1, where the largest spatio-temporal grid corresponds to the bounding box of the action, and adaptive temporal pyramid is used to take into account the variations in motion speed and frequency when different people perform the same action.

3.2.2 Fisher Vector

FV [16] assumes that the generation process of local descriptors $\mathbf{v}_t$ can be modeled by a probability density function $p(.;\theta)$ with parameters θ. In order to describe the contribution of individual parameters to the generative process, one can compute the gradient of the log-likelihood of the data on the model:

$$\mathcal{G}_\theta^{\mathbf{X}} = \frac{1}{T} \nabla_\theta \log p(\mathbf{X}; \theta).$$

The probability density function is usually modeled by Gaussian Mixture Model (GMM), and $\theta = \{w_1, \boldsymbol{\mu}_1, \boldsymbol{\sigma}_1, \ldots, w_K, \boldsymbol{\mu}_K, \boldsymbol{\sigma}_K\}$ are the model parameters of the K-component GMM, where w_k, $\boldsymbol{\mu}_k$ and $\boldsymbol{\sigma}_k$ are respectively the mixture weight, mean vector, and diagonal covariance matrix of Gaussian k. In our work, we use the following formulas [17] to calculate the FV components:

$$\mathcal{G}_{\boldsymbol{\mu}_k}^{\mathbf{X}} = \frac{1}{T\sqrt{w_k}} \sum_{t=1}^{T} \gamma_t(k) \left(\frac{\mathbf{v}_t - \boldsymbol{\mu}_k}{\boldsymbol{\sigma}_k} \right),$$

$$\mathcal{G}_{\boldsymbol{\sigma}_k}^{\mathbf{X}} = \frac{1}{T\sqrt{2w_k}} \sum_{t=1}^{T} \gamma_t(k) \left(\frac{(\mathbf{v}_t - \boldsymbol{\mu}_k)^2}{\boldsymbol{\sigma}_k^2} - 1 \right),$$

where $\gamma_t(k)$ is the soft assignment of $\mathbf{v}_t$ to Gaussian k:

$$\gamma_t(k) = \frac{w_k \mathcal{N}(\mathbf{v}_t; \boldsymbol{\mu}_k, \boldsymbol{\sigma}_k)}{\sum_{i=1}^{K} w_i \mathcal{N}(\mathbf{v}_t; \boldsymbol{\mu}_i, \boldsymbol{\sigma}_i)},$$

where $\mathcal{N}(\mathbf{v}; \boldsymbol{\mu}_k, \boldsymbol{\sigma}_k)$ are M-dimensional Gaussian distributions. The FV representation of a sequence is the concatenation of $\mathcal{G}_{\boldsymbol{\mu}_k}^{\mathbf{X}}$ and $\mathcal{G}_{\boldsymbol{\sigma}_k}^{\mathbf{X}}$. Since $\mathcal{G}_{\boldsymbol{\mu}_k}^{\mathbf{X}}$ and $\mathcal{G}_{\boldsymbol{\sigma}_k}^{\mathbf{X}}$ are M-dimensional vectors, our FVs are $2MK$-dimensional vectors.

The GMM parameters can be trained using the Expectation-Maximization (EM) [4] algorithm to optimize a Maximum Likelihood criterion. While training a GMM using EM is often more expensive than training a codebook using SC (if the same number of visual words is used in both cases), the advantage of the FV encoding method is that once the GMM has been learned, the calculation of action descriptors using FV is much faster than that using the method in Section 3.2.1.

We apply two normalization steps: l_2-normalization and power normalization [19], as they have been shown [17] to significantly improve the recognition accuracy when the FV is combined with a linear classifier. In the power normalization step, the FV is transformed using the function of the form: $z \leftarrow sign(z)|z|^{\rho}$ with $0 < \rho \leq 1$. In our experiments, ρ is set to $\rho = \frac{1}{2}$. In order to further introduce the spatial geometry and temporal order of a depth sequence, we use the spatio-temporal grids given in Fig. 1. We compute the FV representation for each grid, and concatenate the FVs of all the grids to obtain the final representation of the depth sequence.

3.3 Action Recognition

Using the two methods presented in Sections 3.2.1 and 3.2.2 to encode local descriptors into a global representation of a depth sequence, we obtain two algorithms termed SN4D-SC and SN4D-FV. For both the algorithms, we rely on a linear SVM trained in an one-versus-all fashion to build a multi-class classifier for action recognition. In our experiments, we use the LIBLINEAR library [6] that provides the implementation of linear SVMs.

4 Experiments

In this section, we evaluate SN4D-SC and SN4D-FV on two benchmark datasets: MSRAction3D [11] and MSRGesture3D [24], and compare them against several

Method	Accuracy
Bag of 3D Points [11]	74.70
HOJ3D [28]	79.00
EigenJoints [30]	82.30
Random Occupancy Pattern [24]	86.50
Actionlet Ensemble [25]	88.20
Depth Motion Maps Based HOG [31]	88.73
HON4D [15]	88.89
DSTIP [27]	89.30
Pose Set [23]	90.00
Moving Pose [32]	91.70
SNV [29]	93.09
SN4D-SC	**95.45**
SN4D-FV	**91.07**

Table 1: Recognition accuracy comparison of our methods and previous approaches on MSRAction3D.

state-of-the-art methods. For SN4D-SC, the number of visual words was set to $K = 100$. For SN4D-FV, the number of GMM components K was set to $K = 50$.

4.1 MSRAction3D Dataset

The MSRAction3D is an action dataset captured using a depth sensor similar to Kinect. It contains 20 actions performed by 10 different subjects. Each subject performs every action two or three times.

For a fair comparison, we used the experimental setting described in [11]. We divided the 20 actions into three sub- sets AS1, AS2 and AS3, each having 8 actions. Action recognition was performed on each subset separately. We followed the cross-subject test setting, in which subjects 1,3,5,7,9 were used for training and subjects 2,4,6,8,10 were used for testing.

Tab. 1 shows the accuracy of the proposed methods and different state-of-the-art methods. As can be observed, SN4D-SC outperforms the other methods, including HON4D and SNV which also use surface normals in 4D space of depth, time, and spatial coordinates to form local descriptors. The accuracy of SN4D-

Method	AS1	AS2	AS3	Ave.
Depth Motion Maps [3]	96.2	83.2	92.0	90.47
Histogram of Oriented Displacements [7]	92.39	90.18	91.43	91.26
Point in a Lie group [22]	95.29	83.87	**98.22**	92.46
Hierarchical Recurrent Neural Network [5]	93.33	94.64	95.50	94.49
DMM-LBP-FF [2]	98.1	92.0	94.6	94.9
DMM-LBP-DF [2]	**99.1**	92.9	92.0	94.7
SN4D-SC	95.28	**94.69**	97.32	**95.45**
SN4D-FV	91.51	87.96	93.75	91.07

Table 2: Recognition accuracy comparison of our methods and previous approaches on AS1, AS2 and AS3 of MSRAction3D.

SC, SN4D-FV and some state-of-the-art methods on AS1, AS2, AS3 is given in Tab. 2. SN4D-SC achieves the highest average accuracy. The confusion matrices are shown in Fig.2. For both SN4D-SC and SN4D-FV, most of the confusion is observed between actions that share similar arm motion like *tennis serve* and *pick up & throw* (AS1 and AS3), *draw circle* and *high arm wave* (AS2).

4.2 MSRGesture3D Dataset

The MSRGesture3D dataset contains 12 dynamic hand gestures defined by the American sign language. Each gesture is performed two or three times by 10 subjects.

For a fair comparison, we used the leave-one-subject-out cross validation scheme proposed by [24]. Tab. 3 shows the accuracy of our methods and some state-of-the-art methods. We can observe that SN4D-FV achieves the highest accuracy. SN4D-SC also competes with SNV and outperforms the remaining methods. The confusion matrices are shown in Fig. 3. For SN4D-SC, the most confusion occurs between the actions *finish* and *bathroom*. For SN4D-FC, the most confusion occurs between the actions *finish* and *milk*.

4.3 Effect of Second-Order Partial Derivatives in The Local Descriptor

In this section, we show the effectiveness of $\frac{\partial^2 z}{\partial x^2}$ and $\frac{\partial^2 z}{\partial y^2}$ in the proposed local descriptor by comparing SN4D-SC and SN4D-FV against their variants termed

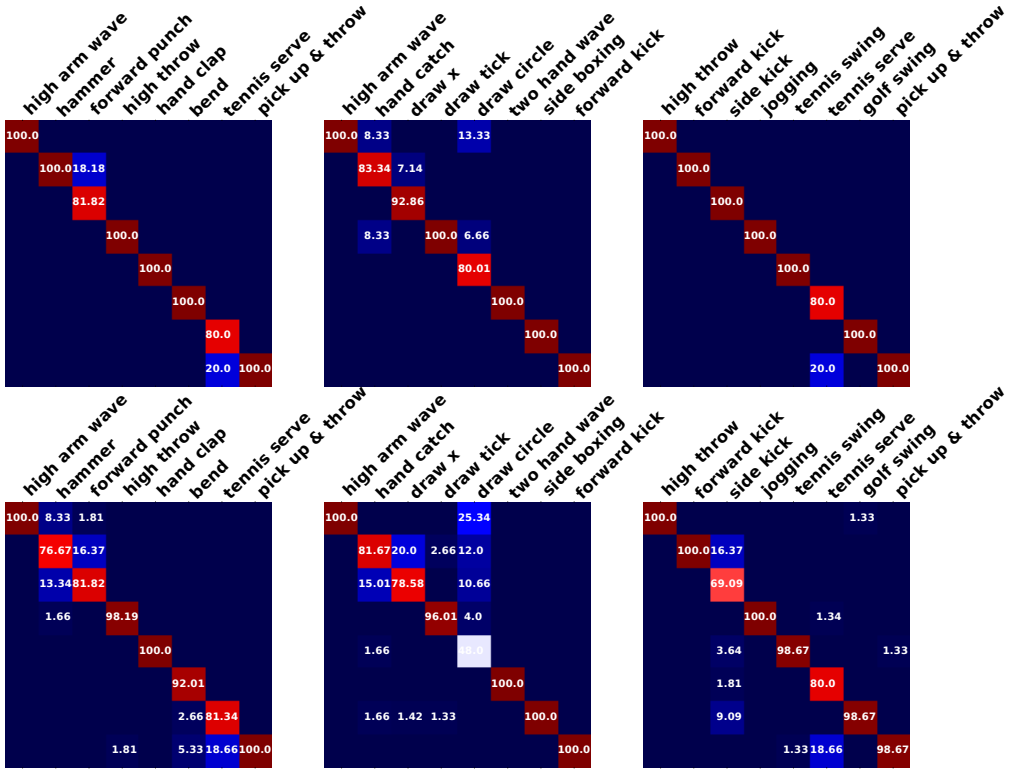

Figure 2: (Best viewed in color) The confusion matrices of our methods on MSRAction3D (top: SN4D-SC, bottom: SN4D-FV, left: AS1, middle: AS2, right: AS3).

SN4D-SC$_v$ and SN4D-FV$_v$, where the components $\frac{\partial^2 z}{\partial x^2}$ and $\frac{\partial^2 z}{\partial y^2}$ are removed from the local descriptor. Tab. 4 shows the accuracy of the four methods. We can observe that in terms of average accuracy over the two datasets, SN4D-SC performs slightly better than SN4D-SC$_v$ and SN4D-FV outperforms SN4D-FV$_v$. This result shows that second-order partial derivatives of depth values along x and y axes carry useful shape cues for improving the recognition accuracy.

5 Conclusion

In this paper, we proposed a new local descriptor for human action recognition in depth images which jointly captures the shape and motion cues using surface

Method	Accuracy
Action Graph on Silhouette [8]	87.70
Random Occupancy Pattern [24]	88.50
Depth Motion Maps Based HOG [31]	89.20
HON4D [15]	92.45
SNV [29]	94.74
SN4D-SC	**94.2**
SN4D-FV	**95.03**

Table 3: Recognition accuracy comparison of our methods and previous approaches on MSRGesture3D.

	MSRAction3D	MSRGesture3D	Ave.
SN4D-SC$_v$	94.04	**94.28**	94.16
SN4D-FV$_v$	86.39	94.85	90.62
SN4D-SC	**95.45**	94.2	**94.82**
SN4D-FV	**91.07**	**95.03**	**93.05**

Table 4: Effect of second-order partial derivatives in our local descriptor.

normals in 4D space of depth, time, spatial coordinates and second-order partial derivatives of depth values along spatial coordinates. Using SC and FV for feature encoding, we obtain high-dimensional action descriptors which are then fed into a linear SVM for efficient classification. We experimentally showed that the proposed methods are effective and more accurate than many state-of-the-art methods on two benchmark datasets.

References

[1] O. Boiman, E. Shechtman, and M. Irani. In Defense of Nearest-Neighbor Based Image Classification. In *CVPR*, pages 1–8, 2008.

[2] C. Chen, R. Jafari, and N. Kehtarnavaz. Action Recognition from Depth Sequences Using Depth Motion Maps-based Local Binary Patterns. In *WACV*, pages 1092–1099, 2015.

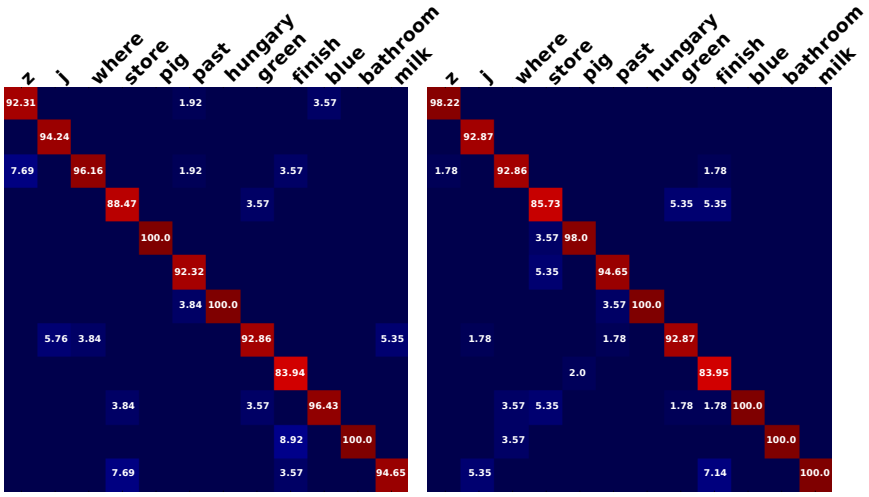

Figure 3: (Best viewed in color) The confusion matrices of our methods on MSRGesture3D (left: SN4D-SC, right: SN4D-FV).

[3] C. Chen, K. Liu, and N. Kehtarnavaz. Real-time Human Action Recognition Based on Depth Motion Maps. *Journal of Real-Time Image Processing*, pages 1–9, 2013.

[4] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of The Royal Statistical Society*, 39(1):1–38, 1977.

[5] Y. Du, W. Wang, and L. Wang. Hierarchical Recurrent Neural Network for Skeleton Based Action Recognition. In *CVPR*, pages 1110–1118, 2015.

[6] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. LIBLINEAR: A Library for Large Linear Classification. *Journal of Machine Learning Research*, 9:1871–1874, 2008.

[7] M. A. Gowayyed, M. Torki, M. E. Hussein, and M. El-Saban. Histogram of Oriented Displacements (HOD): Describing Trajectories of Human Joints for Action Recognition. In *IJCAI*, pages 1351–1357, 2013.

[8] A. Kurakin, Z. Zhang, and Z. Liu. A Real-Time System for Dynamic Hand Gesture Recognition with A Depth Sensor. In *EUSIPCO*, pages 1975–1979, 2012.

[9] H. Lee, A. Battle, R. Raina, and A. Y. Ng. Efficient Sparse Coding Algorithms. In *NIPS*, pages 801–808. 2007.

[10] W. Li, Z. Zhang, and Z. Liu. Expandable Data-Driven Graphical Modeling of Human Actions Based on Salient Postures. *IEEE Transactions on Circuits and Systems for Video Technology*, 18(11):1499–1510, 2008.

[11] W. Li, Z. Zhang, and Z. Liu. Action Recognition Based on A Bag of 3D Points. In *CVPRW*, pages 9–14, 2010.

[12] K. Mikolajczyk and C. Schmid. A Performance Evaluation of Local Descriptors. *TPAMI*, 27(10):1615–1630, 2005.

[13] M. Müller. *Information Retrieval for Music and Motion*. Springer-Verlag New York, Inc., 2007.

[14] R. M. Murray, S. S. Sastry, and L. Zexiang. *A Mathematical Introduction to Robotic Manipulation*. CRC Press, Inc., 1994.

[15] O. Oreifej and Z. Liu. HON4D: Histogram of Oriented 4D Normals for Activity Recognition from Depth Sequences. In *CVPR*, pages 716–723, 2013.

[16] F. Perronnin and C. Dance. Fisher Kernels on Visual Vocabularies for Image Categorization. In *CVPR*, pages 1–8, 2007.

[17] F. Perronnin, J. Sanchez, and T. Mensink. Improving the Fisher Kernel for Large-scale Image Classification. In *ECCV*, pages 143–156, 2010.

[18] L. R. Rabiner. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[19] J. Sanchez, F. Perronnin, T. Mensink, and J. Verbeek. Image Classification with the Fisher Vector: Theory and Practice. *IJCV*, 105(3):222–245, 2013.

[20] J. Schmidhuber. Deep Learning in Neural Networks: An Overview. *Neural Networks*, 61:85–117, 2015.

[21] S. Tang, X. Wang, X. Lv, T. X. Han, J. Keller, Z. He, M. Skubic, and S. Lao. Histogram of Oriented Normal Vectors for Object Recognition with a Depth Sensor. In *ACCV*, pages 525–538, 2013.

[22] R. Vemulapalli, F. Arrate, and R. Chellappa. Human Action Recognition by Representing 3D Skeletons as Points in a Lie Group. In *CVPR*, pages 588–595, 2014.

[23] C. Wang, Y. Wang, and A. L. Yuille. An Approach to Pose-Based Action Recognition. In *CVPR*, pages 915–922, 2013.

[24] J. Wang, Z. Liu, J. Chorowski, Z. Chen, and Y. Wu. Robust 3D Action Recognition with Random Occupancy Patterns. In *ECCV*, pages 872–885, 2012.

[25] J. Wang, Z. Liu, Y. Wu, and J. Yuan. Mining Actionlet Ensemble for Action Recognition with Depth Cameras. In *CVPR*, pages 1290–1297, June 2012.

[26] L. Wang and D.-C. He. Texture Classification Using Texture Spectrum. *Pattern Recognition*, 23(8):905–910, 1990.

[27] L. Xia and J. K. Aggarwal. Spatio-temporal Depth Cuboid Similarity Feature for Activity Recognition Using Depth Camera. In *CVPR*, pages 2834–2841, 2013.

[28] L. Xia, C. C. Chen, and J. K. Aggarwal. View Invariant Human Action Recognition Using Histograms of 3D Joints. In *CVPRW*, pages 20–27, 2012.

[29] X. Yang and Y. Tian. Super Normal Vector for Activity Recognition Using Depth Sequences. In *CVPR*, pages 804–811, 2014.

[30] X. Yang and Y. L. Tian. EigenJoints-based Action Recognition Using Naive-Bayes-Nearest-Neighbor. In *CVPRW*, pages 14–19, 2012.

[31] X. Yang, C. Zhang, and Y. Tian. Recognizing Actions Using Depth Motion Maps-based Histograms of Oriented Gradients. In *Proceedings of the 20th ACM International Conference on Multimedia*, pages 1057–1060, 2012.

[32] M. Zanfir, M. Leordeanu, and C. Sminchisescu. The Moving Pose: An Efficient 3D Kinematics Descriptor for Low-Latency Action Recognition and Detection. In *ICCV*, pages 2752–2759, 2013.

[33] Y. Zhu, W. Chen, and G. Guo. Fusing Spatiotemporal Features and Joints for 3D Action Recognition. In *CVPRW*, pages 486–491, 2013.